

SLA 1101



6

①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Übersetzung der
europäischen Patentschrift

⑧7 EP 0 358 292 B 1

⑩ DE 689 28 311 T 2

⑤1 Int. Cl.⁶:
G 06 F 9/445
G 06 F 15/16
G 06 F 9/44

- ②1 Deutsches Aktenzeichen: 689 28 311.3
- ⑧6 Europäisches Aktenzeichen: 89 302 132.9
- ⑧6 Europäischer Anmeldetag: 3. 3. 89
- ⑧7 Erstveröffentlichung durch das EPA: 14. 3. 90
- ⑧7 Veröffentlichungstag
der Patenterteilung beim EPA: 10. 9. 97
- ④7 Veröffentlichungstag im Patentblatt: 2. 4. 98

DE 689 28 311 T 2

- ③0 Unionspriorität:
240955 06. 09. 88 US
- ⑦3 Patentinhaber:
Digital Equipment Corp., Maynard, Mass., US
- ⑦4 Vertreter:
Betten & Resch, 80469 München
- ⑧4 Benannte Vertragstaaten:
DE, FR, GB, NL

- ⑦2 Erfinder:
Flaherty, James E., Hudson Massachusetts 01749,
US

⑤4 Fern-Urlader

DE 689 28 311 T 2

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

89 302 132.9

HINTERGRUND DER ERFINDUNG

Zu einem Computer gehören sowohl die als Hardware bezeichneten physikalisch vorhandenen Geräte als auch die Software, die Befehle, die das physikalisch vorhandene Gerät steuern. Die Software unterteilt sich in Anwender- und Systemprogramme. Zur Anwendungssoftware zählen solche Programme, die ausschließlich dazu bestimmt sind, spezifische, durch den Anwender gestellte Aufgaben zu lösen. Programme, die die Hardware des Computers sowie die Ausführung der Anwenderprogramme steuern, zählen demgegenüber zur Systemsoftware. Darüber hinaus zählt zur Systemsoftware das Betriebssystem, d. h. das Programm, das den eigentlichen Computer oder Prozessor (CPU) sowie die Peripheriegerätetreiber steuert, die ihrerseits die Ein- und Ausgabegeräte (I/O) wie Drucker oder Terminals steuern.

Universalcomputer sind relativ komplizierte Geräte. In der Regel warten mehrere Anwenderprogramme in einer Warteschlange darauf, den Prozessor nutzen zu dürfen. Das Betriebssystem muß herausfinden, welches Programm als nächstes läuft, wieviel Prozessorzeit es verwenden darf und auf welche weiteren Betriebsmittel das Programm zugreifen darf. Darüber hinaus benötigt jedes Anwenderprogramm ein spezielles Eingabe- oder Ausgabegerät, wobei das Anwenderprogramm seine Daten an das Betriebssystem, das die Peripheriegerätetreiber steuert, übertragen muß.

In der Regel ist das Betriebssystem zu groß und zu kompliziert, als daß es im nichtflüchtigen Speicher (ROM) gespeichert werden könnte. Darüber hinaus wird das Be-

triebssystem in regelmäßigen Abständen modifiziert, so daß es auch unter diesem Gesichtspunkt unzweckmäßig wäre, es in einem ROM zu speichern. Gewöhnlich wird das Betriebssystem deshalb auf einer Magnetplatte oder Festplatte gespeichert und zur Ausführung in den Direktzugriffsspeicher (RAM) geladen. Das Problem besteht darin, daß der Zugriff auf die Platte den entsprechenden Hardwaretreiber und die restliche System-Software erfordert. Unter diesem Gesichtspunkt scheint überhaupt keine Möglichkeit zu bestehen, den Computer zu starten, da zum Lesen des Betriebssystems von der Platte bereits das Betriebssystem erforderlich ist.

Als Ausweg aus diesem Problem dient ein als Umladen oder Booten bezeichnetes Verfahren. Beim Booten stellt ein spezieller ROM, der sogenannte Boot-ROM, ein kleines Programm zur Verfügung. Wird der Computer neu gestartet, wird der Inhalt des Boot-ROM gelesen und das entsprechende Programm wird ausgeführt. Das Boot-ROM-Programm ist ein kleines Programm, das dem Prozessor mitteilt, wo sich auf der Platte ein umfangreicheres Bootprogramm, das sogenannte "Boot-Block-Programm" befindet. Darüber hinaus teilt es dem Prozessor mit, wie das Programm in den Speicher zu laden und auszuführen ist. Das Boot-Block-Programm wird nun ausgeführt und kopiert das Betriebssystem von der Platte. Befindet sich das Betriebssystem einmal im Speicher, wird es ausgeführt und der Computer ist vollständig betriebsbereit.

Aus mehreren Universalcomputern, sogenannten Knoten, die über eine Kommunikationsverbindung miteinander kommunizieren, kann ein Universalcomputer-Netz aufgebaut werden. Sind die Computer weit voneinander entfernt, spricht man von einem Weitverkehrsnetz (WAN), sind sie nur wenig voneinander entfernt und befinden sich z. B. im gleichen Gebäude, von einem lokalen Netz (LAN). Bei einem bestimm-

ten Typ lokaler Netze ist jeder Computer bzw. Knoten über eine Schnittstelle, eine sogenannte Netzvorrichtung, an die gemeinsame Kommunikationsverbindung angeschlossen.

Die an ein Netz angeschlossenen Computer kommunizieren, indem sie sich gegenseitig Nachrichten zusenden. Jede dieser Nachrichten umfaßt ein Anfangsblockfeld, das einen Identifizierer mit der Adresse der Schnittstelle desjenigen Computers, an den die Nachricht gerichtet ist, und einen Identifizierer mit der Adresse der Schnittstelle desjenigen Computers, der die Nachricht sendet, enthält, sowie ein Datenfeld, das die zwischen den Computern zu übertragenden Informationen enthält.

Sämtliche Netzvorrichtungen verfolgen die Nachrichten auf der Verbindung, kopieren die Nachrichten von der Verbindung in den Speicher des Computers und teilen dem Computer mit, ob der Nachrichtenanfängsblock die Adresse der betreffenden Netzvorrichtung enthält oder ob der Nachrichtenanfängsblock eine Rundschreiben-Adresse enthält, bei der die Nachricht an alle Vorrichtungen des Netzes gehen soll, oder ob der Nachrichtenanfängsblock eine Mehrfach-Adresse enthält, die anzeigt, daß die Nachricht an alle Vorrichtungen innerhalb eines bestimmten Adreßbereichs gehen soll. Möchte einer der Computer eine Nachricht an einen anderen Computer übertragen, fügt er seiner eigenen Adresse die des beabsichtigten Empfängers und beide Angaben wiederum den zu übertragenden Informationen hinzu und erzeugt so eine Nachricht, die über die Kommunikationsverbindung übertragen wird.

Die Verfügbarkeit lokaler Netze hat die Vielseitigkeit der Computer noch erhöht. Es dauerte nicht lange, bis die Anwender erkannten, daß das Netz Dateien auf Festplatten eines an das Netz angeschlossenen Computers allen an das Netz angeschlossenen Computern zur Verfügung stellen

kann. So wurden verschiedene Programme entwickelt, um den Zugriff auf Dateien auf den Festplatten anderer Computersysteme zu vereinfachen. Schließlich entstand die Vorstellung, daß man bestimmte Computer des Netzes mit besonderen Aufgaben betrauen sollte. So kann z. B. einer der Computer jedem über das Netz ansprechbaren physikalischen Gerät einen logischen Namen zuordnen. Auf diese Weise kann ein Anwender, anstatt eine Datei von einer bestimmten Festplatte eines bestimmten Systems anzufordern, die Datei einfach unter Verwendung eines logischen Namens anfordern. Ein Computer des Netzes, der speziell für die entsprechenden Verknüpfungen vorgesehen ist, fordert die Datei dann von der bestimmten Festplatte des betreffenden Systems an, wobei er die Festplatte jenes Systems so behandelt, als wäre es eine lokale Festplatte des Anwenders. Die Übersetzung der logischen Namen in physikalische Geräte erfolgt für den Anwender transparent. Der Computer, der in diesem Fall die Übersetzung übernimmt, wird als Festplatten- oder Dateiserver bezeichnet. Darüber hinaus wurden eine Reihe weiterer Serverfunktionen festgelegt, so die eines Druckservers, der den Druck von Dateien erlaubt, ohne daß angegeben werden muß, an welchen Computer der Drucker angeschlossen ist.

Weiterhin kann man einer Festplatte einen Anwender zuordnen und anschließend diesen Anwender mittels des Festplattenservers mit seiner Festplatte verbinden, unabhängig davon, auf welchem Computer des LAN er arbeitet. Ein als lokale Festplatte (LAD) bezeichnetes Festplattenserver-Programm gestattet dem auf einem bestimmten Computer arbeitenden Anwender weiter, eine Datei auf der Festplatte eines anderen Computers als virtuelle Festplatte zu behandeln. Eine solche virtuelle Festplatte verhält sich so, als befände sie sich auf dem Computer des Anwenders selbst.

Zu den Dateien, auf die über das Netz zugegriffen werden kann, gehören auch diejenigen des Betriebssystems selbst. In der Tat wurden bereits vor der Vereinfachung des Dateizugriffs durch das LAD-Konzept zahlreiche Verfahren entwickelt, um ein Betriebssystem auf einen Computer des Netzes "herunterzuladen". Somit wurde es unnötig, daß der Computer, der das Betriebssystem empfängt, selbst ein solches auf der Festplatte hat oder überhaupt noch über eine Festplatte verfügt. Damit ein solches Verfahren funktioniert, müssen lediglich zwei Voraussetzungen erfüllt sein. Erstens muß die Netzvorrichtung in der Lage sein, eine Nachricht zu generieren und über die Kommunikationsverbindung zu übertragen, um so das Betriebssystem von einem anderen Computer des Netzes anzufordern. Zweitens muß die Netzvorrichtung in der Lage sein, das einmal empfangene Betriebssystem in den Speicher zu laden und dafür zu sorgen, daß die CPU dieses ausführt.

Bei einer früheren Anordnung verschickte die Netzvorrichtung beim Einschalten des Computers eine Boot-Anforderung. Diese zu allen anderen Computern des Netzes übertragene Aufforderung umfaßt eine einfache Nachricht mit der Hardware-Adresse der Netzvorrichtung. Ein an das Netz angeschlossener Computer sucht bei Empfang dieser Anforderung in einer Datenbank nach einem Eintrag, der dem anfordernden Computer ein Betriebssystem zuweist. Findet der empfangende Computer in seiner Datenbank einen Eintrag für den anfordernden Computer, wird er zum Host-Computer, liest das angeforderte Betriebssystem von seiner Festplatte, fügt der Datei die Adresse des anfordernden Computers hinzu, und überträgt die Nachricht an das Netz.

Der anfordernde Computer empfängt die Nachricht, kopiert sie in seinen Speicher und beginnt mit der Ausführung des

Betriebssystems. Obgleich der Vorgang des Ladens eines Betriebssystems in einen Computer eines Netzes im Prinzip einfach zu verstehen ist, ist die tatsächliche Ausführung recht kompliziert.

Genauer führt der anfordernde Computer beim Einschalten zunächst ein Selbsttest- und Einschaltsequenz aus. Zu dieser Sequenz gehört, daß der Prozessor nach einem Bootgerät sucht. Schlägt die Suche nach einem solchen Bootgerät fehl, gestattet der Prozessor der Netzvorrichtung, ein Booten über das Netz anzufordern.

In der Regel greift die Netzvorrichtung dabei nicht auf die sonst bei der Kommunikation zwischen Computern üblichen Protokolle zurück. Das ist deshalb so, weil solche Protokolle komplizierter sind, als es diese einfache Aufgabe erfordert; die Schnittstelle müßte hierfür wesentlich mehr Funktionalität unterstützen, als für die Aufgabe tatsächlich benötigt wird. Statt dessen wird ein einfaches Protokoll eingesetzt, das aus einer kleinen Menge spezialisierter Nachrichten zum Betrieb, zur Prüfung, für Bootanforderungen usw. besteht (vgl. z. B. DECnet^T Digital Network Architecture Maintenance Operations Functional Specification, Best.-Nr. AA-x436A-TK DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS). So kann die Bootanforderungsnachricht zum Beispiel einfach die Codenummer für eine Bootanforderung, die Codenummer der des Typs der anfordernden Vorrichtung sowie den Typ des angeforderten Computers und Informationen darüber, ob es sich bei dem angeforderten Programm um ein Urladeprogramm oder das Betriebssystem handelt, enthalten. Diesen Codes fügt die Netzvorrichtung des anfordernden Computers noch ihre Adresse sowie eine weitere Adresse, die die Nachricht als Rundschreiben an alle Computer des Netzes kennzeichnet, zu.

Sobald ein weiterer Computer des Netzes die Nachricht empfängt, entnimmt er dieser die Hardwareadresse und durchsucht die Liste der Computer, für die er Betriebssysteme besitzt. Findet ein Computer, der die Nachricht empfangen hat, die Hardware-Adresse des anfordernden Computers nicht in seiner Datenbank, ignoriert er die Nachricht einfach. Reagiert innerhalb eines bestimmten Zeitraums keiner der anderen Computer auf die Nachricht des anfordernden Computers, sendet der anfordernde Computer eine erneute Bootanforderung. Findet der empfangende Computer die Adresse des anfordernden Computers dagegen in seiner Datenbank, überprüft er, ob das Betriebssystem des anfordernden Systems sofort oder schrittweise geladen werden soll.

Das Problem bei diesem Verfahren, ein Betriebssystem über das Netz zu laden, besteht darin, daß bereits vor der Bootanforderung zu viel Informationen im Netz vorliegen müssen. Das heißt, ein Computerknoten des Netzes muß bereits zuvor wissen, welche Programme einschließlich Bootprogramme und Betriebssysteme jeder bootbare Netzcomputer des Netzes benötigen könnte. Damit das Betriebssystem mit den Boot-Programmen zusammenarbeitet, muß es außerdem so angepaßt werden, daß es "Hooks" genannte Einspringpunkte enthält, die die Bootprogramme aufrufen können.

ZUSAMMENFASSUNG DER ERFINDUNG

Die Erfindung wie in Anspruch 1 beansprucht umfaßt ein neues und verbessertes Verfahren zum Herunterladen von Betriebssystemen oder anderen ausführbaren Programmen auf einen Computer in einem Netz, wobei der Computer über keine Bootvorrichtung verfügt und die herunterladbare Abbildung nicht modifiziert werden muß, und, wie in

Anspruch 11 beansprucht, einen Knoten zum Anschluß an ein System.

Beim Einschalten fordert die Netzvorrichtung ein Booten und über das Netz wird ein Minimum-Boot-Programm zum Speicher des anfordernden Computers übertragen. Dieses Fern-Minimum-Boot-Programm stellt die zum Aufrechterhalten einer Netzverbindung zum Festplattenserver benötigte Funktionalität zur Verfügung. Durch Verknüpfen mit dem Einschalt- und Selbsttest-Programm des anfordernden Computers veranlaßt das Fern-Minimum-Boot-Programm den Computer, so zu arbeiten, als ob er über eine lokale Festplatte mit einer Abbildung des Betriebssystems verfügen würde, während tatsächlich alle Festplatten-Zugriffe des Computers über das Netz übertragen werden. Da das Netz wie eine Festplatte behandelt wird, entsteht keine Notwendigkeit, das Betriebssystem beim Herunterladen zu modifizieren.

KURZBESCHREIBUNG DER ZEICHNUNGEN

Die Erfindung ist in den beigegeführten Ansprüchen im einzelnen dargelegt. Die obenerwähnten sowie weitere Vorteile dieser Erfindung lassen sich besser verstehen unter Bezugnahme auf die folgende Beschreibung in Verbindung mit den beigegeführten Abbildungen, in denen

Fig. 1 ein Blockschaltbild eines in Übereinstimmung mit der Erfindung konstruierten Systems zeigt, das einen Host-Computer sowie einen zu bootenden Knoten umfaßt;

Fig. 2 ein Blockschaltbild der Struktur des Minimum-Boot-Programms darstellt;

Fig. 3 ein Schrittschema des Bootvorgangs zeigt.

BESCHREIBUNG EINER ERLÄUTERNDEN AUSFÜHRUNGSFORM

Fig. 1 zeigt ein Netz, das aus einem Computer-Knoten 10 ohne lokale Bootvorrichtung sowie einem Host-Computer-Knoten 14 mit einer Festplatte 20, die das Betriebssystem des Computer-Knotens 10 enthält, besteht, die beide durch eine Kommunikationsverbindung 12 verbunden sind. Der Knoten 10 verfügt über eine Kommunikationsschnittstelle, die Netzvorrichtung 26 genannt wird, mit deren Hilfe er auf der Kommunikationsverbindung 12 Nachrichten sendet und empfängt. Analog besitzt der Knoten 14 eine Kommunikationsschnittstelle 28.

Der Knoten 10 verfügt über zahlreiche Elemente eines gewöhnlichen Computers, etwa über einen Speicher 27 und einen Prozessor (CPU). Darüber hinaus kann er Ein- und Ausgabegeräte wie ein Datensichtgerät VDT 16 und einen lokalen Festplattenspeicher 29 besitzen.

Auch der Host-Computer 14 verfügt über die Elemente jedes Computers wie CPU 30 und Speicher 32 sowie Ein- und Ausgabegeräte wie etwa ein Datensichtgerät VDT 18 und eine Festplatte 20. Die Festplatte 20 des Host-Computers 14 enthält nicht nur das Betriebssystem des Host-Computers 23 selbst, sondern auch ein Minimum-Boot-Programm 34, das erforderlich ist, um den Knoten 10 zu laden sowie eine Datei 33 einer virtuellen Festplatte, die das Betriebssystem 22 des Knotens 10 enthält.

Bei Einschalten führt der Prozessor 24 des Computers 10 eine schrittweise Einschalt-/Selbsttestsequenz aus, in dessen Verlauf er versucht, eine Bootvorrichtung mit einer Datei zu finden, die sein Betriebssystem enthält. Findet er keine solche Bootvorrichtung, sendet die Netzvorrichtung 26 ein Rundschreiben mit der Betriebssystem-Anforderung über die Netz-Kommunikationsverbindung 12 ab.

Die Schnittstelle 28 des Host-Computers 14 stellt fest, daß es sich bei der Nachricht um ein Rundschreiben handelt, und benachrichtigt die CPU 30. Die CPU 30 des Host-Computers 14 stellt fest, daß es sich bei der Nachricht um eine Bootanforderung handelt und versucht, in ihrem Speicher 32 ein Betriebssystem für den Knoten 10 zu finden. Gelingt es ihr, ein solches zu identifizieren, konstruiert sie eine Nachricht, indem sie einer Minimum-Boot-Programm-Datei 34 die Hardwareadresse der Schnittstelle 26 des Knoten-Computers 10 hinzufügt und die Nachricht an die Kommunikationsverbindung 12 übergibt.

Die Kommunikationsvorrichtung 26 des Knotens 10 empfängt die Nachricht und kopiert das Minimum-Boot-Programm in den unteren Speicher-Adreßbereich 36 des Speichers 27 dieses Knotens. Danach beginnt die CPU 24, das Minimum-Boot-Programm auszuführen. Anschließend wird das Minimum-Boot-Programm in den oberen Adreßbereich 46 des Speichers 27 kopiert und die Einschalt-/Selbsttestsequenz ist abgeschlossen.

Hierbei behandelt die Software des Knotens 10 die Datei 33 der Festplatte 20 des Hosts 14 im wesentlichen wie ihre virtuelle lokale Festplatte. Der Knoten 10 liest einen Boot-Block 56 der auf der Festplatte 20 befindlichen Datei 33 einer virtuellen Festplatte und benutzt diesen zum Laden des Betriebssystems 22. Dabei muß besonders darauf hingewiesen werden, daß all dies über die Netzverbindung 12 und nicht durch Zugriff auf den lokalen Festplattenspeicher 29 des Knotens 10 erfolgt.

Um die Abfolge der Vorgänge im Detail zu verstehen, muß zunächst der Charakter des Minimum-Boot-Programms betrachtet werden. Das in Fig. 2 dargestellte Minimum-Boot-Programm besteht aus einem Anfangsblock 41, der es für das Betriebssystem des Host-Computers als Netzprogramm

ausweist. Als nächstes folgt ein Bootsteuerprogramm 42, das den Knoten 10 auf das Herunterladen vorbereitet. Der als lokale Festplatte (LAD) bezeichnete Abschnitt 48 ist ein Programm, mit dessen Hilfe der Knoten 10 Dateien auf einer Festplatte des Netzes als virtuelle lokale Festplatte behandeln kann. Ein lokaler Systemtransport (LAST) 50 stellt die eigentlichen Kommunikationsfunktionen des lokalen Netzes zur Verfügung, wie sie in einer gleichzeitig anhängigen Anmeldung von Bruce E. Mann u. a., Aktenzeichen des Anwalts No. PD 88-0421 mit dem Titel "LOCAL AREA SYSTEM TRANSPORT" beschrieben sind. Ein Sicherungsschicht-Protokoll (DLL) 52 dient als Kommunikationsmechanismus zwischen den Knoten über die Netzvorrichtung 26 des Knotens 10 und die Netzvorrichtung 28 des Knotens 14. Ein Ablaufsteuerprogramm 54 dient schließlich zur Echtzeitabwicklung und Zeitüberwachung der Netzkommunikation.

Fig. 3 zeigt die durch den Knoten 10 und den Host 14 während des Herunterladens des Betriebssystems für den Knoten 10 ausgeführten Schritte im Detail. Auf der linken Seite stehen die von dem Knoten 10, auf der rechten die von dem Host 14 ausgeführten Schritte. Die Abbildungsmitte zeigt die zwischen beiden über die Kommunikationsverbindung 12 übertragenen Daten. Die Pfeilrichtung zeigt die Richtung des Nachrichtenflusses an.

Wie in den Fig. 1, 2 und 3 gezeigt ist, führt der Computer 10 dann, wenn er eingeschaltet wird, einen Einschalt-/Selbsttest (Schritt 100) aus. Im Rahmen des Einschalt-/Selbsttests überprüft die CPU 24 verschiedene Bus-Adressen, um zu ermitteln, ob das System über eine Erweiterungskarte verfügt. Wird eine solche Erweiterungskarte gefunden, werden irgendwelche auf der Karte befindlichen Initialisierungsroutinen abgearbeitet. Handelt es sich bei der Erweiterungskarte um eine Netzvorrichtung, for-

dern die in der Vorrichtung befindlichen Initialisierungsroutinen (Schritt 102) durch Absetzen eines Rundschreibens mit einer Bootanforderung (Schritt 104) ein Booten über das Netz an. Ein weiterer mit dem Netz verbundener Computer 14 empfängt (Schritt 106) die Bootanforderung und durchsucht (Schritt 108) seine Datenbank nach Informationen über den anfordernden Knoten 10. Ist der anfordernde Knoten 10 in der Datenbank enthalten, wird die jeweils geeignete Minimum-Boot-Routine von der Festplatte 20 gelesen (Schritt 110) und an den anfordernden Knoten 10 übertragen (Schritt 112).

Wenn der Knoten 10 die Minimum-Boot-Routine empfängt (114), kopiert er diese in den unteren Speicherbereich 36 und führt sie - insbesondere das Bootsteuerprogramm 42 - aus (Schritt 116). Das Minimum-Bootsteuer-Programm 42 der Minimum-Boot-Routine ermittelt zunächst (Schritt 118), über wieviel Speicher 27 der Knoten 10 verfügt. In einer Ausführungsform, bei der der Knoten 10 einen IBM-kompatiblen Personalcomputer umfaßt, geschieht dies durch Aufrufen einer Interrupt-12-Interrupt-Dienstroutine. Die Bootsteuerung 42 modifiziert (Schritt 120) die gemeldete Speichergröße, indem sie die Größe des Minimum-Boot-Programms von der durch den Interrupt 12 in einem bestimmten Speicherbereich abgelegten Speichergröße subtrahiert (Fig. 2). Dies erfolgt, damit der Knoten 10 auf diesen Speicherbereich nicht zugreifen kann. Die Minimum-Boot-Routinen werden in den oberen Speicherbereich 46 kopiert (Schritt 122), und die Boot-Steuerung wird mit einer Routine verknüpft (Schritt 124), die das Lesen eines Bootblocks von einer lokalen Festplatte ermöglicht. Auf diese Weise kann die Einschalt-/Selbsttestsequenz zum Abschluß kommen und seine restlichen Schritte selbst steuern.

Die Einschalt-/Selbsttestsequenz übernimmt so wieder die Steuerung (Schritt 126) der restlichen Schritte der Initialisierungsroutinen der Netzvorrichtung 26 und führt diese bis zum Schluß aus (Schritt 128). Abschließend versucht die Einschalt-/Selbsttestsequenz (Schritt 130), den Bootblock der lokalen Festplatte zu lesen. Bei der Ausführung mit einem IBM-kompatiblen Personalcomputer als Knoten 10 geschieht dies durch Initialisieren der Interrupt-19-Dienstroutine. Hierdurch übernimmt die Boot-Steuerroutine 42 wieder die Steuerung (Schritt 132). Die Boot-Steuerroutine 42 ruft (Schritt 134) das LAD-Programm 48, das LAST-Programm 50, das DLL-Programm 52 und das Ablaufsteuerprogramm 54 an deren Einspringpunkten auf und initialisiert diese. Bei der Ausführung mit einem IBM-kompatiblen Personalcomputer als Knoten 10 erzeugt jeder Festplattenzugriff einen Interrupt-13. Das LAD-Programm wird mit Interrupt 13 verknüpft, so daß jeder Festplattenzugriff durch das LAD-Programm abgefangen wird (Schritt 136).

Das Bootsteuerprogramm 42 übernimmt wiederum (Schritt 138) die Steuerung und gestattet ein Auslesen (Schritt 140) der Register der Netzvorrichtung 26, um so die Netzadresse des Knotens 10 zu bestimmen. Das Bootsteuerprogramm 42 ruft (Schritt 142) das LAD-Programm 48 auf, um eine Verbindung zur lokalen Festplatte herzustellen. Ist die Verbindung zur lokalen Festplatte des Servers einmal hergestellt (Schritt 144), kann der Knoten auf die virtuelle Festplatte zugreifen. Nachdem die Verbindung zur lokalen Festplatte einmal hergestellt ist, können die Kommunikationsparameter im Grunde geändert werden. So werden zum Beispiel bei einer in den Fig. 3C und 3D gezeigten Ausführung zu diesem Zeitpunkt die DEC-Net Parameter des Knotens 10, etwa eine Netzadresse, durch den Server gelesen (Schritt 146), damit die LAD-Routine 48 am Knoten 10 zum Lesen (Schritt 148) der Parameterda-

tei zur Verfügung steht. Nachdem die Parameter einmal gelesen sind, wird die LAD-Verbindung abgebaut (Schritt 150) und anschließend mit den neuen LAD-Parametern wiederhergestellt (Schritt 152). Wenn, wie es meist der Fall ist, die Kommunikationsparameter nicht geändert zu werden brauchen, können die Schritte 146, 148, 150 und 152 entfallen, und der Knoten 10 fährt mit dem Bootvorgang fort und erteilt eine Leseanforderung, um den Bootsektor 56 zu empfangen.

Das Bootsteuerprogramm 42 gibt eine Leseanforderung an den Bootsektor 56 aus (Schritt 154). Bei der Ausführung mit einem IBM-kompatiblen Personalcomputer als Knoten 10 geschieht dies über den Interrupt-13. Da das LAD-Programm 48 mit der Interruptroutine verkettet ist (Schritt 136), unterbricht es den Aufruf und leitet ihn zum Host 14 (Schritt 156). Dieser Host 14 liest (Schritt 158) den Bootsektor 56 auf der Festplatte 20 und gestattet dem Knoten 10, ihn ebenfalls zu lesen (Schritt 160). Das auf dem Knoten 10 laufende LAD-Programm tauscht dann Nachrichten mit dem Host 14 aus (Schritt 162), um die Übertragung des Bootsektors 56 auf den Kasten 10 und dessen Ausführung (Schritt 164) zu erleichtern.

Mit Hilfe des Bootsektors 56 kann der Knoten 10 das Betriebssystem anfordern, indem er den Interrupt-13 auslöst (Schritt 166), der wiederum vom LAD 48 abgefangen und zum Server geleitet wird (168). Der Server liest (Schritt 170) das Betriebssystem aus der auf der Festplatte 20 befindlichen Datei 33 und stellt es dem LAD 48 des Knotens 10 zur Verfügung (Schritt 172). Das LAD-Programm 48 gestattet dem Knoten 10, das Betriebssystem 22 zu lesen (Schritt 174) und in den Speicher 27 zu laden. Ist das gesamte Betriebssystem geladen, wird das LAD-Programm vom Interrupt 13 abgekoppelt (Schritt 176) und auf die lokale Festplatte 29 zurückgegeben. Danach

wird die Steuerung an den Initialisierungscode des ausführbaren Programms oder des Betriebssystems übertragen (Schritt 178).

Es sollte darauf hingewiesen werden, daß bei dieser Ausführungsform der Host und der Server durch den gleichen Knoten 14 bezeichnet sind. Dies ist jedoch nicht zwingend erforderlich, und es sind andere Ausführungsform möglich, bei denen beide getrennt sind.

Weiter muß darauf hingewiesen werden, daß keiner der Schritte beim Herunterladen des Betriebssystems irgendwelche Modifikationen des Betriebssystems erforderte. Da die Host-Festplatte wie eine lokale Festplatte behandelt wird, kann jede ausführbare Abbildung ohne Modifikation geladen werden.

Anhand der Darstellung der bevorzugten Ausführungsform wird der Fachmann die Möglichkeit zahlreicher im Rahmen der beanspruchten Erfindung liegender Varianten erkennen. Es besteht deshalb die Absicht, die Erfindung lediglich durch den Umfang der Ansprüche einzugrenzen.

Ansprüche

1. Verfahren zum Herunterladen einer Abbildung eines ausführbaren Programms (executable image) von einem Host-Computer (14) über eine Kommunikationsverbindung (12) zu einem Knoten (10), wobei der Knoten eine Schnittstelle (26) aufweist, über die er über die Kommunikationsverbindung kommuniziert, wobei das Verfahren folgende Schritte aufweist:

a) den Knoten in die Lage zu versetzen, ein Fern-Minimum-Boot-Programm (34) über die Kommunikationsverbindung aufzufinden (104-114) und das Fern-Minimum-Boot-Programm (34) mit einer lokalen Abbildungs-Auffindungs-Interrupt-Dienstroutine zu verbinden;

b) Ausführen (134) des Fern-Minimum-Boot-Programms in Reaktion auf eine lokale Abbildungs-Auffindungs-Interrupt-Dienstaufforderung (130), die durch den Knoten erzeugt wird, um das Auffinden (144, 158, 170) einer Abbildung eines ausführbaren Programms von dem Host zu initiieren;

c) Ausführen (78) der Abbildung eines ausführbaren Programms durch den Knoten,

dadurch gekennzeichnet, daß das Verfahren ferner folgende Schritte umfaßt:

d) Initiieren der Ausführung (100, 102) eines Knoteninitialisierungsprogramms durch den Knoten, wobei das Initialisierungsprogramm die Ausführung eines Schnittstelleninitialisierungsprogrammes initiiert, um die Schnittstelle zu initialisieren;

e) Ausführen des Schnittstellen-Initialisierungsprogrammes durch den Knoten, um den Knoten in die Lage zu versetzen, das Fern-Minimum-Boot-Programm aufzufinden, woraufhin der Knoten die Ausführung des Knoten-Initialisierungsprogramms (102) abschließt.

2. Verfahren nach Anspruch 1, bei dem der Knoten (10) nach dem Auffinden der Abbildung des ausführbaren Programms das Fern-Minimum-Boot-Programm von der lokalen Abbildungs-Auffindungs-Interrupt-Dienstroutine entfernt (176).

3. Verfahren nach Anspruch 1, bei dem die Ausführung des Schnittstelleninitialisierungsprogramms folgende Schritte umfaßt:

a) Feststellen (102), daß die Schnittstelle (26) dazu konfiguriert ist, das Fern-Minimum-Boot-Programm über die Kommunikationsverbindung (12) zu empfangen; und

b) Anfordern (104) des Fern-Minimum-Boot-Programms (34) über die Kommunikationsverbindung.

4. Verfahren nach Anspruch 3, bei dem Knoten (10) einen Speicher (27) aufweist, der einen Adreßraum aufweist, der sich von einem niedrigen Adreßwert zu einem hohen Adreßwert erstreckt, und bei dem der Knoten nach der Anforderung des Fern-Minimum-Boot-Programms (34) folgendes durchführt:

a) Laden (114) des Fern-Minimum-Boot-Programms in den Speicher an einem Teil (36) des Adreßraums, der nahe an dem niedrigen Adreßwert liegt;

b) Ermitteln (118) der Größe des Speichers an dem Knoten;

c) Laden eines neuen Speichergrößenwerts, der gleich der Speichergröße minus der Größe des Fern-Minimum-Boot-Programms ist, an eine Speichergrößen-Interruptdienst-Speicherstelle; und

d) Kopieren (122) des Fern-Minimum-Boot-Programms in den Speicher an einen Teil (46) des Adreßraums, der nahe bei dem hohen Adreßwert liegt.

5. Verfahren nach Anspruch 4, bei dem das Bestimmen der Speichergröße an dem Knoten folgende Schritte umfaßt:

a) Erzeugen einer Speichergrößen-Interrupt-Dienstanforderung; und

b) Lesen eines an der Speichergrößen-Interruptdienst-Speicherstelle zurückgelieferten Wertes.

6. Verfahren nach Anspruch 1, bei dem das Erzeugen (130) der lokalen Abbildungs-Auffindungs-Interrupt-Dienstanforderung in Reaktion auf die vollständige Ausführung (128) des Knoten-Initialisierungsprogramms geschieht.

7. Verfahren nach Anspruch 1, bei dem die Ausführung (134) des Fern-Minimum-Boot-Programms folgende Schritte umfaßt:

a) Binden (136-142) eines lokalen Speicher-Simulationsprogramms, welches eine Speichereinrichtung auf dem Host-Computer als eine lokale Speichereinrichtung behandelt, durch den Knoten (10) mit einer lokalen Speichereinrichtungs-Interrupt-Dienstroutine;

b) Herstellen (144) der Kommunikation mit dem Host durch den Knoten; und

c) Lesen (146) der in einer Speichereinrichtung auf dem Host befindlichen Abbildung des ausführbaren Programms durch den Knoten.

8. Verfahren nach Anspruch 7, bei dem die Herstellung (144) der Kommunikation mit dem Host durch den Knoten folgende Schritte umfaßt:

a) Lesen eines Adreßregisters der Schnittstelle, um die Adresse des Knotens zu bestimmen;

b) Erzeugen einer lokalen Speichereinrichtungs-Interrupt-Dienstanforderung;

c) Abfangen der lokalen Speichereinrichtungs-Interrupt-Dienstanforderung während der Verarbeitung des lokalen Speicher-Simulationsprogramms; und

d) Verwenden der Adresse während der Verarbeitung des lokalen Speicher-Simulationsprogramms, um eine Nachricht an den Host zu senden, um eine Verbindung mit der Host-Speichereinrichtung herzustellen.

9. Verfahren nach Anspruch 7, bei dem der Knoten nach der Ausführung des Fern-Minimum-Boot-Programms das lokale Speicher-Simulationsprogramm von der lokalen Speichereinrichtungs-Interrupt-Dienstroutine entfernt (150).

10. Verfahren nach Anspruch 1, bei dem die Ausführung des Schnittstelleninitialisierungsprogramms folgende Schritte umfaßt:

i) Ausgeben einer Boot-Anforderung durch den Knoten über die Kommunikationsleitung;

ii) Laden des Fern-Minimum-Boot-Programms in einen niedrigen Speicherbereich des Knotens durch den Host;

iii) Ausführung des Fern-Minimum-Boot-Programms durch den Knoten;

iv) Kopieren des Fern-Minimum-Boot-Programms in den hohen Speicherbereich des Knotens;

v) Verbinden des Fern-Minimum-Boot-Programms mit einer Interrupt-Dienstroutine des Knotens;

vi) Ausführen eines System-Selbsttests durch den Knoten;

wobei die Ausführung des Fern-Minimum-Boot-Programms folgende Schritte umfaßt:

i) Zurückgabe der Kontrolle des Fern-Minimum-Boot-Programms durch den System-Selbsttest;

ii) Initialisieren einer Vielzahl von Funktionen durch den Knoten;

iii) Herstellen einer Verbindung zwischen dem Knoten und dem Host;

iv) Lesen eines Boot-Sektors auf dem Host durch den Knoten;

v) Laden der Abbildung des ausführbaren Programms;

vi) Freigeben der LAD(Local Area Disk)-Interruptbearbeitung.

11. Knoten (10) zum Anschluß in einem System, welches einen Host-Computer (14), eine Kommunikationsverbindung (12), die den Host mit dem Knoten verbindet, sowie eine Schnittstelle (26), über die der Knoten über die Kommunikationsverbindung kommuniziert, einschließt, wobei der Knoten umfaßt:

a) eine Einrichtung, die den Knoten in die Lage versetzt, ein Fern-Minimum-Boot-Programm (34) über die Kommunikationsverbindung (12) aufzufinden und das Fern-Minimum-Boot-Programm mit einer lokalen Abbildungs-Auffindungs-Interrupt-Dienstroutine zu binden;

b) eine Einrichtung (24), um nach dem Binden des Fern-Minimum-Boot-Programms mit der lokalen Abbildungs-Auffindungs-Interrupt-Dienstroutine und in Reaktion auf eine lokale Abbildungs-Auffindungs-Interrupt-Dienstanforderung das Fern-Minimum-Boot-Programm auszuführen, um das Auffinden einer Abbildung des ausführbaren Programms (22) von dem Host zu starten;

c) eine Einrichtung (24), um die Abbildung des ausführbaren Programms auszuführen, wobei der Knoten dadurch gekennzeichnet ist, daß er ferner einschließt:



d) eine Einrichtung (24), um auf dem Knoten ein Knoten-Initialisierungsprogramm auszuführen, welches ein Schnittstelleninitialisierungsprogramm umfaßt;

e) eine Einrichtung (24) zum Ausführen des Schnittstellen-Initialisierungsprogramms, um den Betrieb der Einrichtung zu ermöglichen, die es dem Knoten ermöglicht, das Fern-Minimum-Boot-Programm abzurufen.

12. Knoten nach Anspruch 11, wobei der Knoten ferner eine Einrichtung zum Entfernen des Fern-Minimum-Boot-Programms von der lokalen Abbildungs-Auffindungs-Interrupt-Dienstroutine nach dem Auffinden des ausführbaren Programms umfaßt.

13. Knoten nach Anspruch 11, bei dem der Prozessor zum Ausführen des Schnittstelleninitialisierungsprogramms folgendes umfaßt:

a) eine Einrichtung, um festzustellen, daß die Schnittstelle konfiguriert ist, das Fern-Minimum-Boot-Programm über die Kommunikationsverbindung zu empfangen; und

b) eine Einrichtung zum Anfordern des Fern-Minimum-Boot-Programms über die Kommunikationsverbindung.

14. Knoten nach Anspruch 12, welcher ferner umfaßt:

a) einen Speicher (27), der einen Adreßraum aufweist, der sich von einem niedrigen Adreßwert zu einem hohen Adreßwert erstreckt.

b) eine Einrichtung zum Laden des Fern-Minimum-Boot-Programms in den Speicher in einem Bereich (36) des Adreßraums, der nahe dem niedrigen Adreßwert liegt, und zwar nach der Anforderung des Fern-Minimum-Boot-Programms;

c) eine Einrichtung zum Bestimmen der Größe des Speichers;

d) eine Einrichtung zum Laden eines neuen Speichergrößenwerts, der gleich der Speichergröße minus der Größe des Fern-Minimum-Boot-Programms ist, an eine Speichergrößen-Interruptdienst-Speicherstelle; und

e) eine Einrichtung zum Kopieren des Fern-Minimum-Boot-Programms in einen Speicherbereich (46) des Adreßraums, der nahe dem hohen Adreßwert liegt.

15. Knoten nach Anspruch 13, bei dem die Einrichtung zum Bestimmen der Speichergröße folgendes umfaßt:

a) eine Einrichtung zum Erzeugen einer Speichergrößen-Interrupt-Dienstanforderung; und

b) eine Einrichtung zum Lesen eines an der Speichergrößen-Interruptdienst-Speicherstelle zurückgegebenen Wertes.

16. Knoten nach Anspruch 11, welcher ferner eine Einrichtung zum Erzeugen der lokalen Abbildungs-Auffindungs-Interrupt-Dienstanforderung in Reaktion auf die vollständige Ausführung des Knoten-Initialisierungsprogramms umfaßt.

17. Knoten nach Anspruch 11, bei dem der Prozessor zum Ausführen des Fern-Minimum-Boot-Programms umfaßt:

a) eine Einrichtung zum Binden eines lokalen Speicher-Simulationsprogramms, welches eine Speichereinrichtung (20) auf dem Host-Computer (14) als eine lokale Speichereinrichtung behandelt, mit einer lokalen Speichereinrichtungs-Interrupt-Dienstroutine;

b) eine Einrichtung (26) zum Herstellen der Kommunikation mit dem Host; und

c) eine Einrichtung zum Lesen der Abbildung des ausführbaren Programms, die sich auf einer Speichereinrichtung auf dem Host befindet.

18. Knoten nach Anspruch 16, bei dem die Einrichtung zum Herstellen der Kommunikation mit dem Host folgendes umfaßt:

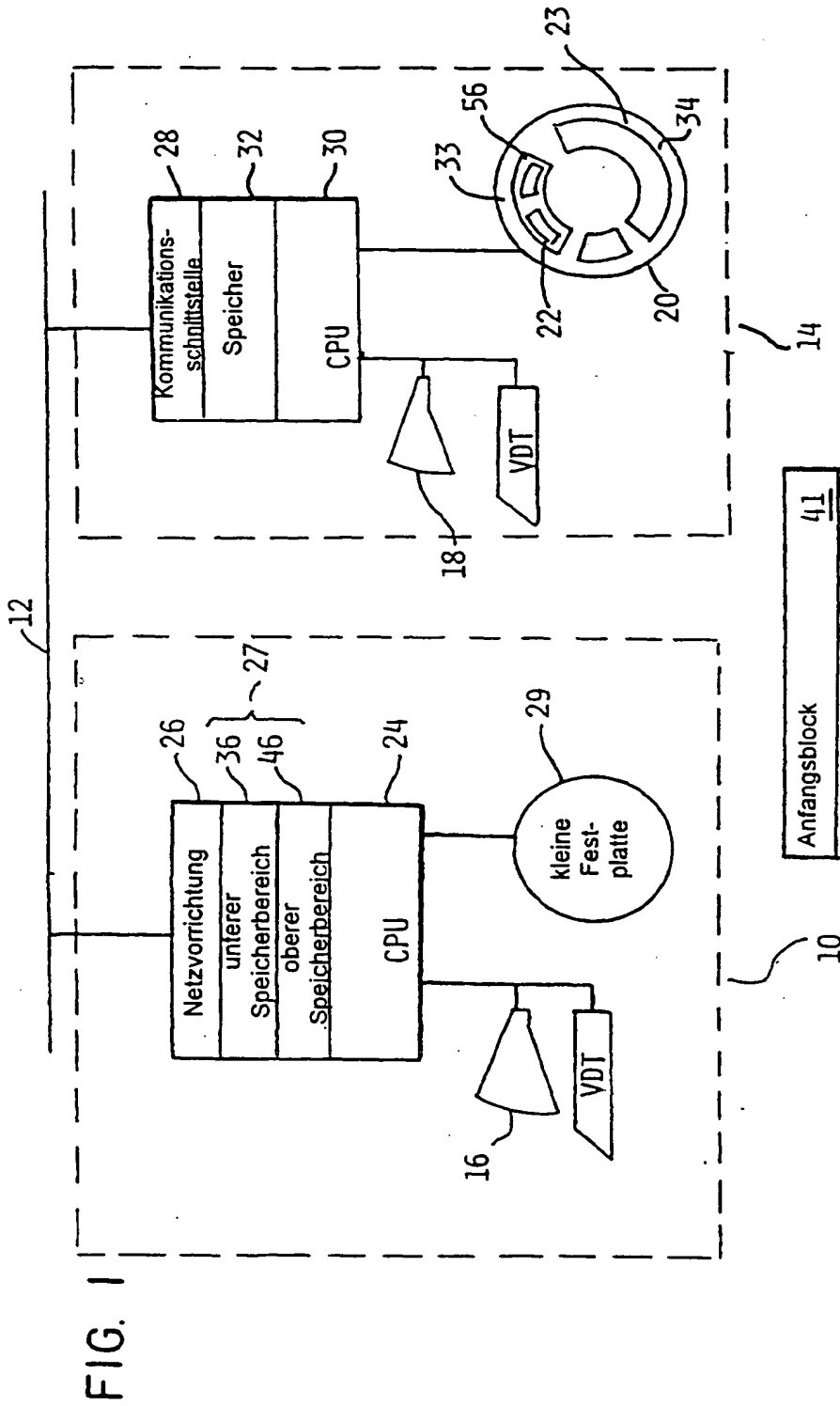
a) eine Einrichtung zum Lesen eines Adreßregisters der Schnittstelle, um die Adresse des Knotens zu bestimmen;

b) eine Einrichtung zum Erzeugen einer lokalen Speichereinrichtungs-Interrupt-Dienstanforderung; und

c) Einrichtung zum Senden einer Nachricht, welche die Adresse des Knotens einschließt, an den Host, um eine Verbindung mit der Host-Speichereinrichtung

herzustellen, in Reaktion auf eine lokalen Speichereinrichtungs-Interrupt-Dienstanforderung.

19. Knoten nach Anspruch 16, welcher ferner eine Einrichtung zum Entfernen des lokalen Speicher-Simulationsprogramms von der lokalen Speichereinrichtungs-Interrupt-Dienstroutine nach der Ausführung des Fern-Minimum-Boot-Programms umfaßt.



Anfangsblock	<u>41</u>
Bootsteuerung	<u>42</u>
LAD	<u>48</u>
LAST	<u>50</u>
DLL	<u>52</u>
Ablaufsteuerprogramm	<u>54</u>

FIG. 2

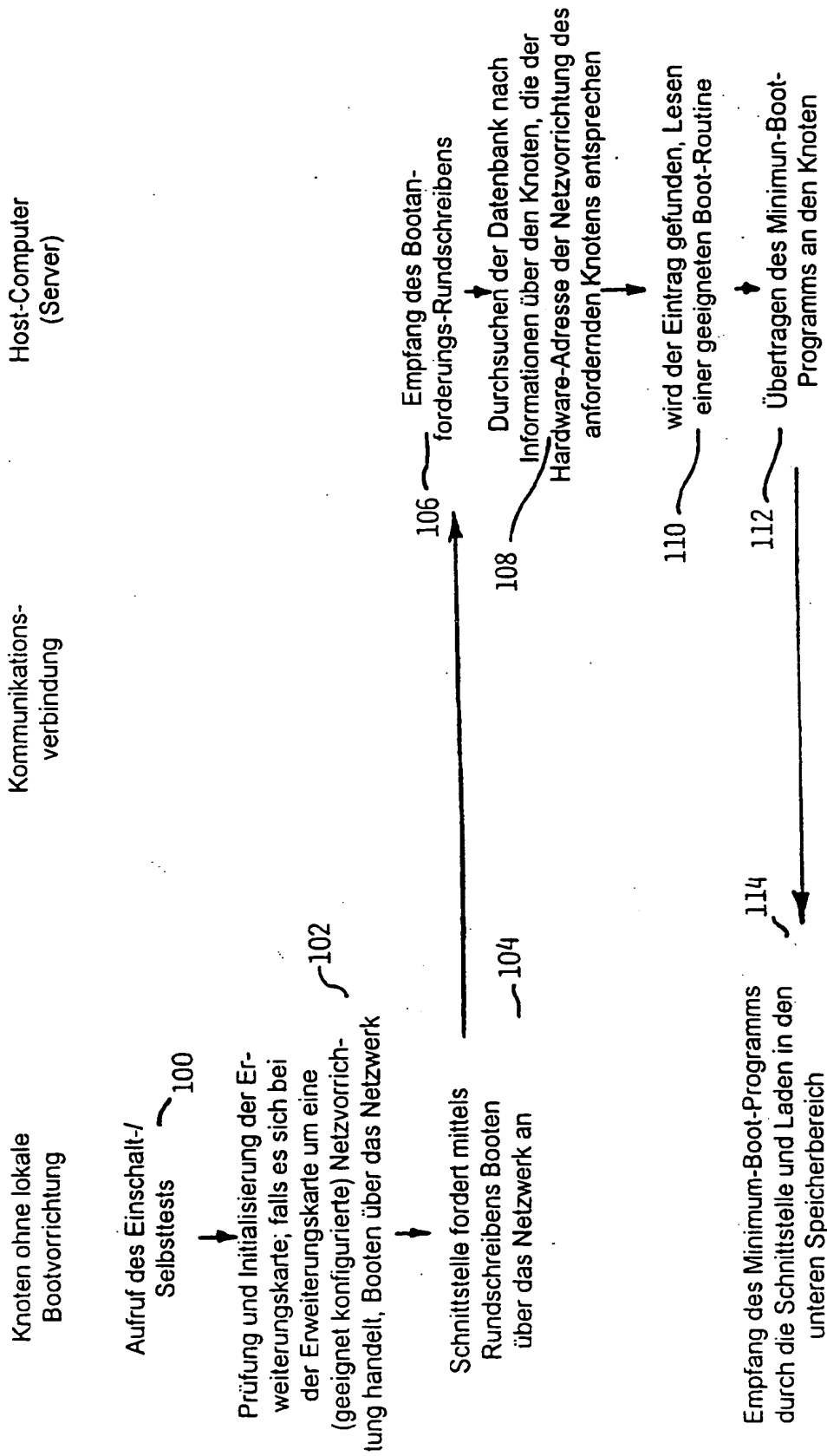


FIG. 3a

(A)

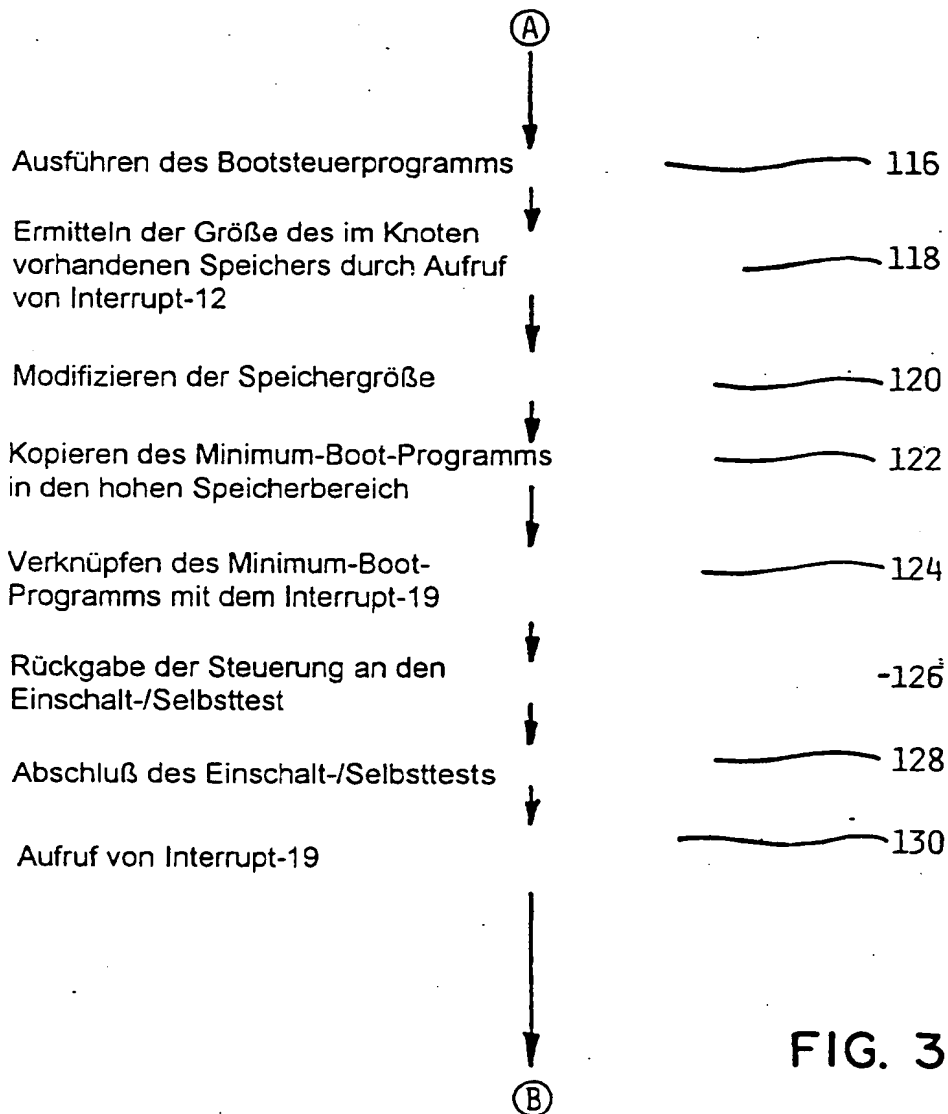


FIG. 3b

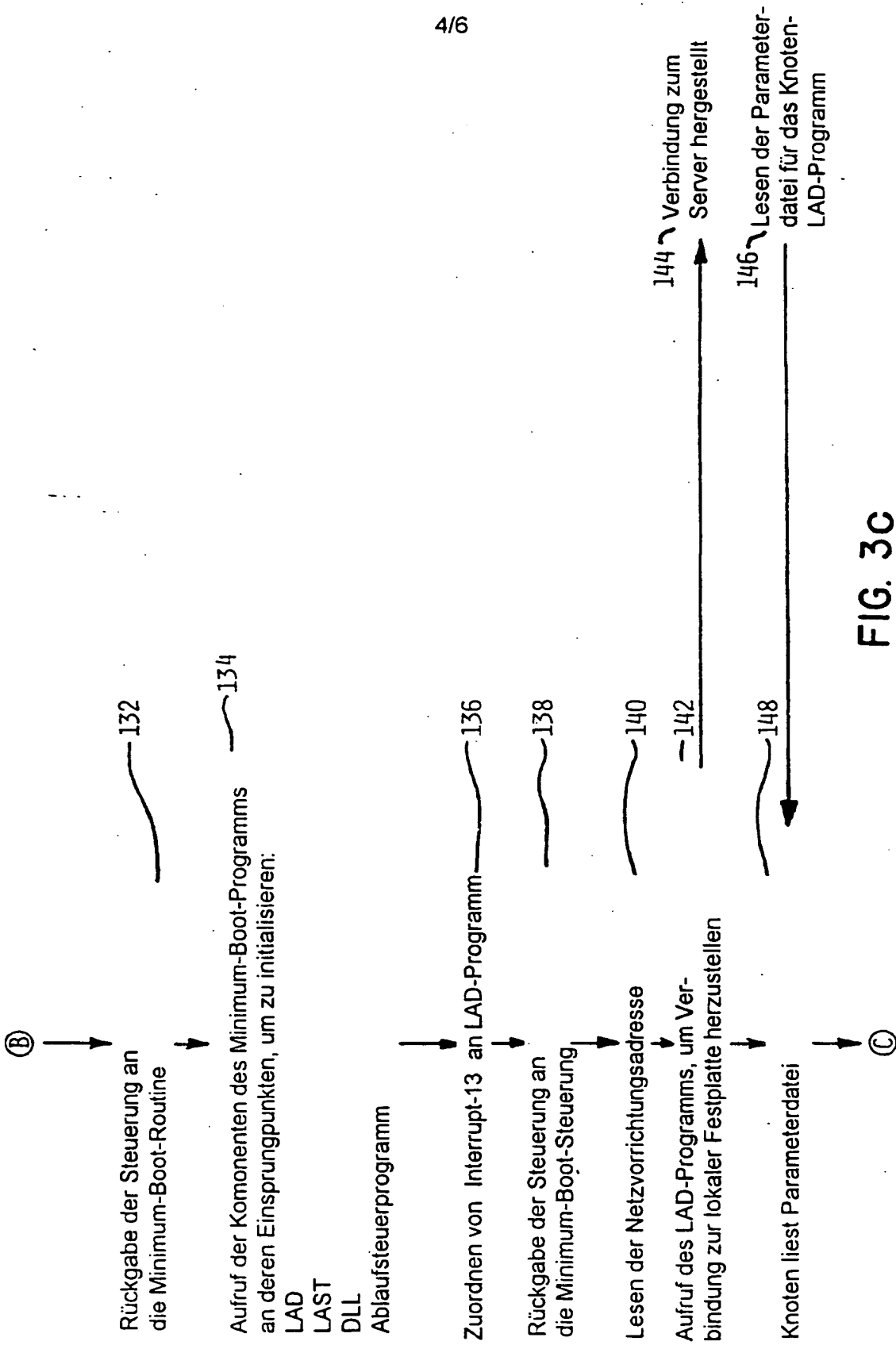


FIG. 3C

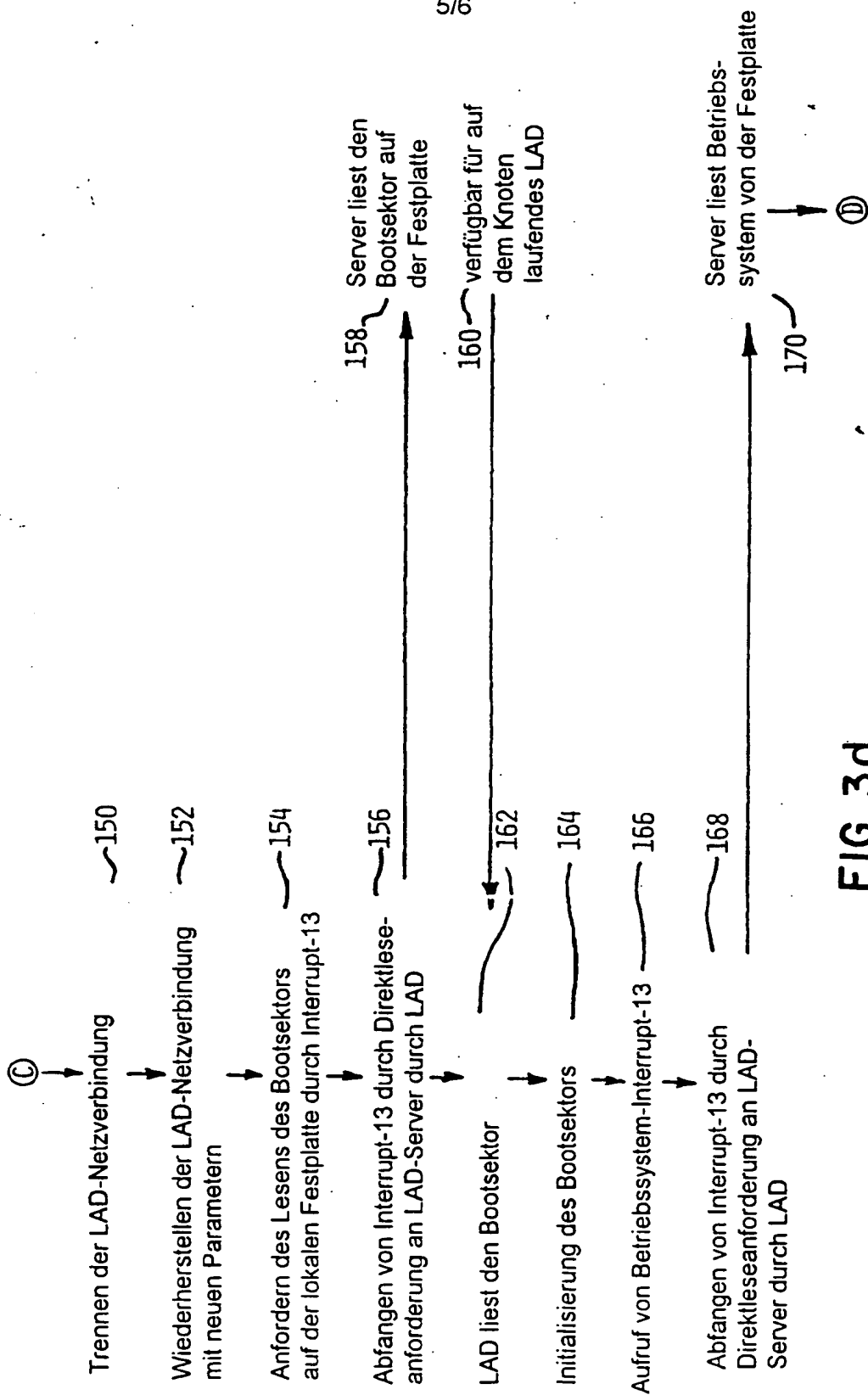


FIG. 3d

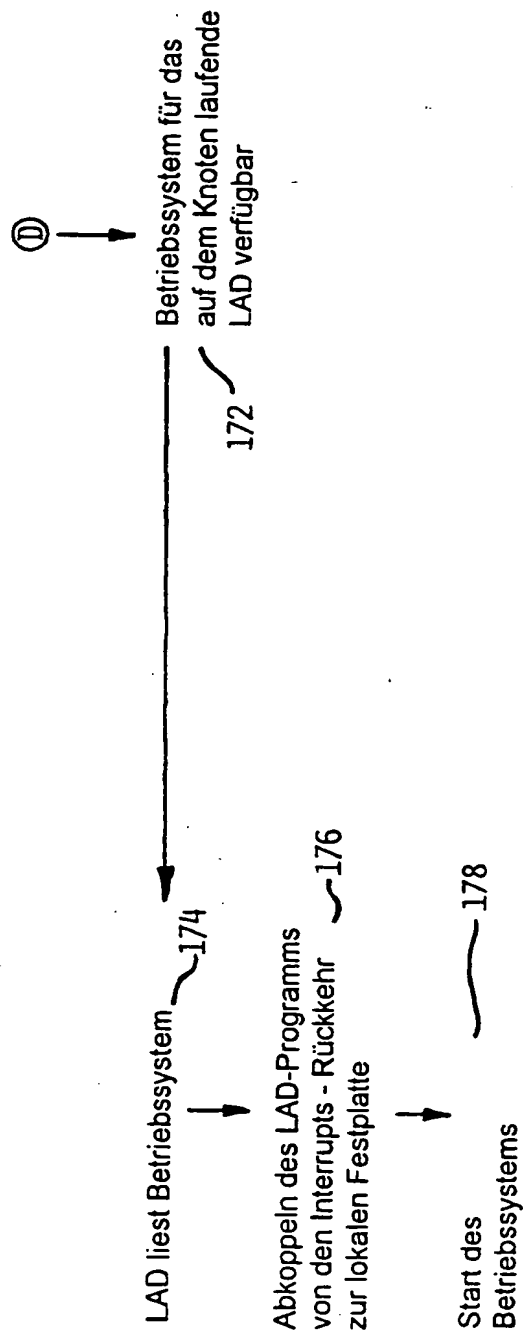


FIG. 3e